The unified Reservoir Computing concept and its digital hardware implementations

David Verstraeten*, Benjamin Schrauwen, Michiel D'Haene and Dirk Stroobandt
Department of Electronics and Information Systems, Ghent University, Ghent, Belgium

1 Reservoir Computing

Recurrent Neural Networks (RNNs) are able by construction to solve temporal problems, such as the control or modelling of non-linear processes. This is due to the recurrent connections, which allow information to be retained for a period of time inside the network. However, this recurrent topology also greatly increases the complexity of the dynamic behaviour of these networks. It is a well known fact that the presence of feedback inside a dynamic system makes it prone to unstable behaviour. This dynamical complexity is for a large part the reason why current learning rules for these types of networks suffer from slow training speeds and convergence problems [9].

Recently, three elegant solutions to this problem have been proposed independently under the terms Liquid State Machine [4], Echo State Machine [1] and Backpropagation Decorrelation (BPDC) [7]. Each of these techniques avoids the training problem of a RNN, while still being able to use its powerful temporal processing capabilities, by using the – randomly chosen – network as a reservoir whose weights are not changed during the training phase. Instead, the response of the reservoir to a certain input is observed from the outside by a conventional and simple classification or regression algorithm that is far easier to train.

These approaches conveniently combine the short-term memory property of the reservoir with the ease of training and fast convergence of a very simple linear readout function. The reservoir acts as a complex kernel-type filter that computes many random nonlinear combinations of the current and past inputs, with a finite memory. The memoryless readout function is then able to linearly combine this information to compute the actual output.

The three methods cited above all take a different approach towards the idea of using a RNN as reservoir. The LSM offers different types of reservoirs, built from very simple nodes like threshold logic gates, to complex and biologically realistic Leaky Integrate & Fire neurons. Echo State Networks are built from sigmoidal neurons and are more directed towards practical applicability. BPDC originates from a very different approach: a mathematically derived training rule for RNNs was found to also lead to the concept of a reservoir, because when applying the rule it appears that only the connections of the output neurons are adjusted.

The combination of our research with results from other groups seems to indicate that these reservoir-based techniques show similarities beyond the simple fact that they all leave a RNN untrained. To substantiate this claim, we applied both an LSM [12] and an ESN [11] to the same isolated speech recognition problem. We found that these two techniques performed quite similarly. Using the LSM-based setup, we were able to attain word error rates (WER) of 0.2% for networks of around 1200 neurons, which is better than a state-of-the-art Hidden Markov Model-based recognizer [12]. Moreover, the small number of parameters describing the ESN allowed us to carry out a rather thorough analysis of the influence of the connection topology. For instance, Figure 1 shows the performance of the ESN as a function of the spectral radius[1] of the connection matrix. This parameter can be interpreted as a measure for the chaotic behaviour of the network. These results indicate that a similar phenomenon to *computation at the edge of chaos*, cited in the context of LSMs [2], also occurs for ESNs despite earlier indications to the contrary. Very recent results for BPDC also seem to confirm this point [8].

We feel that both the structural and the functional similarities between the three techniques cited above means that they can be viewed as different incarnations of the same idea. We therefore propose *Reservoir Computing* as a unifying term.

2 Hardware implementations

It is well known that large neural networks require substantial amounts of computing power and time. This has motivated researchers in the past to investigate the possibility of hardware implementations of these NNs, because the inherent parallel nature of these networks maps very well onto hardware. Our research focuses

*Corresponding author: david.verstraeten@ugent.be.

[1] This is the magnitude of the largest eigenvalue of a matrix

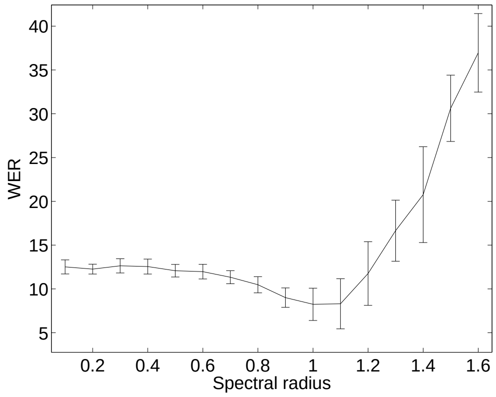

Figure 1: Word Error Rate (lower is better) versus spectral radius with errorbars showing the standard deviation for 20 runs with the same parameter settings.

specifically on compact implementations of spiking neuron models [3] on reconfigurable hardware (Field Programmable Gate Arrays or FPGAs). The binary nature of the spikes and the spiking neural models makes them well suited to be implemented in hardware.

We take different approaches to the implementation: on the one hand we build a timestep-based simulation of directly mapped neurons in hardware, on the other hand we investigate the possibility of performing parallel event-based simulation of very large neural networks. We have now built a powerful and flexibel hardware-framework that enables us to quickly build networks of spiking neurons on reconfigurable hardware [5, 6]. The big advantage of this approach is its flexibility: we can use the same hardware framework to simulate many different types of neurons and network topologies (including reservoirs, see [10]).

3 Conclusion and future work

In this abstract we propose to unify three similar reservoir-based techniques – LSM, ESN and BPDC – under the common term Reservoir Computing. We support this proposal with indications that the reservoirs – despite being built from different nodes or being trained in a different way – show not only structural but also behavioural similarities. We also outline our work around hardware implementations of spiking neural networks.

Our applications up to now include speech recognition and signal generation, but in the near future we plan to apply these techniques to problems from the field of autonomous robotics, and the detection of epileptic attacks both from EEG and non-EEG data (such as small accelerometers attached to the patient's arms and legs).

In [2] several different metrics were introduced and applied to LSMs to try to infer some prior judgement of the computational quality of the reservoirs. We plan to apply these metrics to the other types of reservoirs, hoping to be able to formulate some more general conclusions about the influence of certain parameters on their computational power.

References

[1] H. Jaeger and H. Haas. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, 308:78–80, April 2 2004.

[2] R. Legenstein and W. Maass. *What makes a dynamical system computationally powerful?* MIT Press, 2005.

[3] W. Maass and C. Bishop. *Pulsed Neural Networks*. Bradford Books/MIT Press, Cambridge, MA, 2001.

[4] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[5] B. Schrauwen and M. D'Haene. Compact digital hardware implementations of spiking neural networks. In J. Van Campenhout, editor, *Sixth FirW PhD Symposium*, page on CD, 1 2005.

[6] B. Schrauwen and J. Van Campenhout. Parallel hardware implementation of a broad class of spiking neurons using serial arithmetic. In *Proceedings of ESANN'06*, 2006. To be published.

[7] J. J. Steil. Backpropagation-Decorrelation: Online recurrent learning with O(N) complexity. In *Proceedings of IJCNN '04*, volume 1, pages 843–848, 2004.

[8] J. J. Steil. Stability of backpropagation-decorrelation effi cient O(N) recurrent learning. In *Proceedings of ESANN'05*, Brugge, 2005.

[9] J. Suykens, J. Vandewalle, and B. De Moor. *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. Springer, 1996.

[10] D. Verstraeten, B. Schrauwen, and D. Stroobandt. Reservoir computing with stochastic bitstream neurons. In *Proceedings of the 16th Annual ProRISC Workshop*, pages 454–459, Veldhoven, The Netherlands, 11 2005.

[11] D. Verstraeten, B. Schrauwen, and D. Stroobandt. Reservoir-based techniques for speech recognition. 2006. Submitted to WCCI.

[12] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout. Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, 95(6):521–528, 2005.